Proposing a New Method for Non-relative Imbalanced Dataset

Hamid Parvin, Sara Ansari, and Sajad Parvin

Nourabad Mamasani Branch, Islamic Azad University Nourabad Mamasani, Iran
{hamidparvin,s.ansari}@mamasaniiau.ac.ir

Abstract. A well-known domain in that it is highly likely for each exemplary dataset to be imbalanced is patient detection. In such systems there are many clients while a few of them are patient and the all others are healthy. So it is very common and likely to face an imbalanced dataset in such a system that is to detect a patient from various clients. In a breast cancer detection that is a special case of the mentioned systems, it is tried to discriminate the patient clients from healthy clients. It should be noted that the imbalanced shape of a dataset can be either *relative* or *non-relative*. The imbalanced shape of a dataset is *relative* where the mean number of samples is high in the minority class, but it is very less rather than the number of samples in the majority class. The imbalanced shape of a dataset is *non-relative* where the mean number of samples is low in the minority class. This paper presents an algorithm which is well-suited for and applicable to the field of *non-relative* imbalanced datasets. It is efficient in terms of both of the speed and the efficacy of learning. The experimental results show that the performance of the proposed algorithm outperforms some of the best methods in the literature.

Keywords: Imbalanced Learning, Decision Tree, Artificial Neural Networks, Breast Cancer Detection.

1 Introduction

In fact, each dataset that has an imbalanced distribution among the number of the data points in each of its classes can be considered as an imbalanced dataset. However in artificial intelligence communities, a dataset will be generally considered to be an imbalanced one if only if it has a very high-rated and sharp imbalanced distribution. We call this type of the mentioned imbalanced datasets, the imbalance between classes (e.g. consider the distribution of 10000:100 in a dataset with two classes where one class completely overshadows the other). Of course the imbalance concept is not dependent on the number of classes; it means that it is not only defined for or applicable to the datasets with two classes. It is highly likely that one faces an imbalance dataset having more than two classes. Thus in an imbalanced dataset it is required to use a classifier with a high accuracy in such a way that the minority class

V. Snasel et al. (Eds.): SOCO Models in Industrial & Environmental Appl., AISC 188, pp. 297–306.
springerlink.com

detection is not affected by the majority class detection. It is obvious that the individual evaluation criteria such as overall accuracy or error rate do not provide sufficient information about the quality of learning in an imbalanced dataset.

Imbalanced shape of a dataset is called *intrinsic* where the nature of dataset source involves in being imbalanced. It should be noted that the imbalanced shape of a dataset can be either *relative* or *non-relative*. The imbalanced shape of a dataset is *relative* where the mean number of samples is high in the minority class, but it is very less rather than the number of samples in the majority class. The imbalanced shape of a dataset is *non-relative* where the mean number of samples is low in the minority class. This paper presents an algorithm which is well-suited for and applicable to the field of *non-relative* imbalanced datasets. It is efficient in terms of both of the speed and the efficacy of learning.

2 Backgrounds

A class of solutions to imbalanced datasets tries to apply some changes in dataset to be balanced and then uses a standard learning algorithm. Other class of solutions generally focuses on modifying the standard learning algorithms to be suited and adapted to learn in an imbalanced dataset [5]. In the first approach, there are two common ways: over-sampling and under-sampling. Random over-sampling method takes a set of samples from the minority class and then they are added to dataset. In fact, the number of samples in the minority class is enlarged in such a way that the number of data points in each class, either the minority class or the majority class, gets balanced. Alternatively there is another way to balance an imbalanced dataset named under-sampling method. Unlike the over-sampling method, the under-sampling method reduces a set of samples from the majority class in such a way that the number of data points in each class, either the minority class or the majority class, gets balanced. The over-fitting is the problem that challenges the over-sampling method. The concept losing is the main problem of the under-sampling method. An alternative to overcome the challenges is to turn to informed under-sampling methods. Two of the most well-known methods based on informed under-sampling are *EasyEnsemble* [2] and *BalanceCascade* [3]. Another example of the informed under-sampling methods is based on k-nearest neighbor [4].

In *EasyEnsemble* method it is tried to first produce many classifiers based on different runnings of the under-sampling method, and then to use them as an ensemble of classifiers. It is worthy to note that each mentioned classifier is produced by an AdaBoost mechanism. *EasyEnsemble* is an unsupervised strategy since it uses an independent random sampling with replacement in applying the under-sampling method. *BalanceCascade* method is very similar to *EasyEnsemble* method. *BalanceCascade* explores the sampling in a supervised manner. In *BalanceCascade* method it is tried to iteratively produce a classifier so as to improve the false positive rate of previously produced classifiers.

According to the research findings in the field of imbalanced learning, the criteria employed for assessing the quality of learning of a classifier in an imbalanced dataset are completely different from the common criteria used for evaluating the quality of learning of a classifier in a common dataset. So it is necessary to discuss the evaluation criteria suitable in the field of imbalanced learning. This section explains the approach how to assess the effectiveness of a model in learning of an imbalanced dataset. The common conventional measures to assess a classifier quality in learning of a dataset are the accuracy measure and the error rate measure. These criteria are used for a simple description of a learner (classifier) performance on a dataset but they are not suitable for imbalanced datasets.

Fig. 1 depicts the confusion matrix. In the confusion matrix the *True Positives* are the data points in dataset that have been assigned by classifier to the minority class (the patient class) and they really belong to the minority class. The *False Positives* are the data points in dataset that have been assigned by classifier to the minority class while they really belong to the majority class (the healthy class). The *False Negatives* are the data points in dataset that have been assigned by classifier to the majority class while they really belong to the minority class. The *True Negatives* are the data points in dataset that have been assigned by classifier to the majority class and they really belong to the majority class.

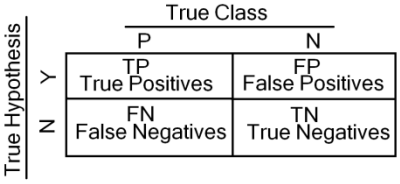

Fig. 1. Confusion Matrix

The performance criteria defined on the imbalanced datasets should be based on the mentioned confusion matrix to be unbiased to the majority class. Studying the confusion matrix makes it clear that the first column shows the number of positive samples (the number of samples in the minority class) and second column shows the number of negative samples (the number of samples in the majority class). It is also clear that the first row shows the number of the samples that classifier recognizes them as the minority class and the second row shows the number of the samples that classifier recognizes them as the majority class. Columns show the distribution of class samples. Indeed each metric using them simultaneously can't be free of sensitivity to class imbalancement. For example accuracy uses both columns and so it is sensitive to imbalancement, i.e. by changing the distribution of the number of data points of the classes of dataset the metric changes while the performance does not change. Some measures which are adjusted for evaluating the learning quality of a classifier at an imbalanced dataset are: (imbalanced) accuracy, precision, recall, F-measure and G-mean [1]. The accuracy of a classifier at an imbalanced dataset is obtained by equation 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

where *TP*, *TN*, *FP* and *FN* stand respectively for the number of *True Positives*, the number of *True Negatives*, the number of *False Positives* and the number of *False Negatives*. The precision is obtained by equation 2.

$$precision = \frac{TP}{TP + FP} \tag{2}$$

where *TP* and *FP* are the same as equation 1. The recall is obtained by equation 3.

$$recall = \frac{TP}{TP + FN} \tag{3}$$

where *TP* and *FN* are the same as equation 1. The F-measure is obtained by equation 4.

$$F - measure = 2 * \frac{precision * recall}{precision + recall} \tag{4}$$

Evaluation based on receiver operating characteristic (ROC) curves, uses two criteria of the two single columns, TP rate and FP rate, of the Fig. 1 and draws a graph depicting the TP rate in terms of the FP rate. ROC curve is a powerful method to evaluate the performance of a learner visually. In precision-recall chart, one could get more information on the performance assessment of a learner [1]. These charts can be considered as the best way to display the performance of a learner in an imbalanced application.

3 Proposed Method

The structure of the proposed algorithm is similar to *EasyEnsemble*. The proposed algorithm initially takes a number of sub-samplings from the majority class with the size of the minority class. Considering each of the sub-sampled data from the majority class in addition to the data of the minority class as a temporal dataset, a decision tree or a multilayer perceptron is trained over the temporal dataset. Finally, all classifiers jointly work as an ensemble. Pseudo code of the proposed ModifiedBagging algorithm is presented in Fig. 2. Like Bagging, Boosting is another meta-algorithm in data mining that is more capable of learning hard problems. The main idea behind Boosting like Bagging is to learn a problem by a set of weak learners and then to create a single strong learner. A weak learner is defined to be a classifier which is only slightly correlated with the true classification or labels; it can label examples better than random guessing. In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification [12]. To complete our conclusion, the ModifiedBoosting is proposed based on the main Boosting algorithm proposed by Schapire [12]. Pseudo code of the proposed ModifiedBoosting is presented in Fig. 3. The proposed ModifiedBoosting is just like the proposed ModifiedBagging, except the majority class is subsampled based on a policy that the error-prone examples have more chance in subsequent samplings.

The ModifiedBagging algorithm pseudo code.

1. Input: A set of minority class examples $\mathcal{S}_{\mathcal{Y}|}$, a set of majority class examples $\mathcal{S}_{\mathcal{Y}-\mathcal{S}}$, $\clubsuit\mathcal{S}_{\mathcal{Y}|}\clubsuit < \clubsuit\mathcal{S}_{\mathcal{Y}-\mathcal{S}}\clubsuit$, the number of subsets, $\mathcal{I}$ to sample from $\mathcal{S}_{\mathcal{Y}-\mathcal{S}}$
2. $\mathcal{i} \Rightarrow 0$
3. **repeat**
4. $\mathcal{i} \Rightarrow \mathcal{i} + 1$
5. Randomly sample a subset $\mathcal{E}\mathcal{i}$ from $\mathcal{S}_{\mathcal{Y}-\mathcal{S}}$, $\clubsuit\mathcal{E}\mathcal{i}\clubsuit = \clubsuit\mathcal{S}_{\mathcal{Y}|}\clubsuit \Leftrightarrow \mathcal{S}\mathcal{J} \mathcal{S}_{\mathcal{Y}|} \cup \mathcal{E}\mathcal{J}$
6. Learn $\mathcal{C}\mathcal{i}$ on $\mathcal{S}\mathcal{i}$. $\mathcal{C}\mathcal{i}$ is a simple classifier
7. **until** $\mathcal{i} = \mathcal{I}$
8. Output: An ensemble $\{C_i|1\ i\ T\}$

Fig. 2. Pseudo code of the proposed ModifiedBagging

Although, as it was mentioned previously, there are many algorithms to deal with learning at imbalanced datasets, this paper only focuses to handle under-sampling approaches. In this group of algorithms, the two of the best algorithms are considered to be *BalanceCascade* and *EasyEnsemble*. It is worthy to be mentioned that the second is one example of informed under-sampling methods [2-3]. As it has been shown [3], these two algorithms absolutely dominate other methods. Their superiority is in terms of both learning efficiency and training speed.

The ModifiedBoosting algorithm pseudo code.

9. Input: A set of minority class examples $\mathcal{S}_{\mathcal{Y}|}$, a set of majority class examples $\mathcal{S}_{\mathcal{Y}-\mathcal{S}}$, $\clubsuit\mathcal{S}_{\mathcal{Y}|}\clubsuit < \clubsuit\mathcal{S}_{\mathcal{Y}-\mathcal{S}}\clubsuit$, the number of subsets, $\mathcal{I}$ to sample from $\mathcal{S}_{\mathcal{Y}-\mathcal{S}}$
10. $\mathcal{i} \Rightarrow 0$
11. W(j)=1, $\forall$ j$\in$[1,T] "S_{min}
12. **repeat**
13. $\mathcal{i} \Rightarrow \mathcal{i} + 1$
14. P(j)=W(j)/sum(W), $\forall$ j$\in$[1,T]
15. Using P randomly sample a subset $\mathcal{E}\mathcal{i}$ from $\mathcal{S}_{\mathcal{Y}-\mathcal{S}}$, $\clubsuit\mathcal{E}\mathcal{i}\clubsuit = \clubsuit\mathcal{S}_{\mathcal{Y}|}\clubsuit \Leftrightarrow \mathcal{S}\mathcal{J} \mathcal{S}_{\mathcal{Y}|} \cup \mathcal{E}\mathcal{J}$
16. Learn $\mathcal{C}\mathcal{i}$ on $\mathcal{S}\mathcal{i}$. $\mathcal{C}\mathcal{i}$ is a simple classifier
17. Test($\{C_i|1\ i\ T\}$, S_{max})
18. W(j)=W(j)*2, $\forall$ j that is misclassified by ensemble $\{C_i|1\ i\ T\}$
19. W(j)=W(j)/2, $\forall$ j that is misclassified by ensemble $\{C_i|1\ i\ T\}$
20. **until** $\mathcal{i} = \mathcal{I}$
21. Output: An ensemble $\{C_i|1\ i\ T\}$

Fig. 3. Pseudo code of the proposed ModifiedBoosting

On the other hand the algorithms of *BalanceCascade* and *EasyEnsemble* are very similar to the proposed algorithm. Therefore, since the two algorithms in terms of the

structure are very similar to the proposed algorithm, and they also dominate other methods, the proposed method is compared to only these two methods in this paper.

The difference between proposed algorithm and *EasyEnsemble* is the main reason of its superiority. The difference is hidden in section 6 of the pseudo code. *EasyEnsemble* uses an AdaBoost classifier ensemble as learner [5]. Using a complex classification system similar to AdaBoost ensemble, not only causes a lot of overhead time for learning, but actually it lacks any justification. It is because after producing classifiers, C_i, voting mechanism is employed. So there is no justification for hierarchically voting in classification part, especially when the minority class has very little data points and hierarchical voting causes sub-sampling from the minority class; it means that hierarchical voting causes to lose the concepts of the minority class. So it is highly likely that the classifiers are not trained properly in the AdaBoost ensemble algorithm due to the small number of samples in the minority class.

The difference between the proposed algorithm and *BalanceCascade* is even more obvious. All differences between the proposed algorithm and *EasyEnsemble* mentioned in the previous section are also differences between the proposed algorithm and *BalanceCascade*. There are also some new differences between the proposed algorithm and *BalanceCascade*. For example, *BalanceCascade* tries to iteratively produce an AdaBoost so as to improve the *FP* of previously produced classifiers. It is again highly likely that the classifiers do not train properly in the AdaBoost algorithm due to the small number of samples in the minority class.

4 Experimental Results

This section evaluates the results of applying the proposed framework on a real imbalanced dataset of breast cancer patients. Dataset has been collected from some real clients of Bidgol-Aran city's hospital [6]. Dataset includes 386 clients. While 17 cases have breast cancer, the rest 369 cases have been healthy. 26 features extracted from each client that the most of them almost belong to the nominal ones. The nominal features to be used in any MLP are first converted to numerical features. It means that if feature A has 4 distinct values, say, $<A_1, A_2, A_3, A_4>$, we consider values A_1, A_2, A_3, A_4 respectively equal to 1, 2, 3, 4. After the coding phase, each feature is normalized into interval [0-1] just for usages in MLPs. The normalizing relations can be calculated by equation 5.

$$nf_{x,i} = \frac{f_{x,i}}{\max\limits_{y}(f_{y,i}) - \min\limits_{y}(f_{y,i})} \tag{5}$$

where $f_{x,i}$ stands for ith feature of xth data point and $nf_{x,i}$ stands for ith normalized feature of xth data point. In the paper, multilayer perceptron and decision tree are used as base classifier. We use multilayer perceptrons with 2 hidden layers including respectively 10 and 5 neurons in the hidden layer 1 and 2, as the base simple classifier. All of decision trees have used in this research employ Gini criterion as decision tree evaluation metric. Parameter Gini criterion for decision tree is set to two.

The classifiers' parameters are kept fixed during all experiments. It is important to be mentioned that type of all classifiers in the algorithms are kept fixed to either only

decision tree or only multilayer perceptron. It means that all classifiers are considered as multilayer perceptron in the first experiments. After that the same experiments are taken by substituting all multilayer perceptrons with decision trees.

To find out how a classifier has learned over the mentioned imbalanced dataset, we always use *leave-one-out cross-validation* technique. First four columns of Table 1 show the quality of learning of different simple classification methods over the mentioned imbalanced dataset by *leave-one-out cross-validation* method in terms of different evaluation measures. As it is inferred from Table 1, although the accuracies of simple decision tree classifier and multilayer perceptron neural network classifier are very high, they do not have good performances at all. This is not something unexpected, because these classifiers assign each queried sample to the majority class. Consequently they hit very high accuracies. While their accuracies are good, they are unable to recognize patients. If one looks at Table 1, it will be clearly identified that the performances of the same classifiers enclosed in the proposed framework are significantly increased; while they still have satisfactory accuracies. As expected, using the decision tree as the base classifier can improve considerably the performance rather than using the multilayer perceptron as the base classifier.

Table 1. Performances of different simple methods obtained by leave-one-out method. MBG and MBT stand for ModifiedBaGging, ModifiedBoosTing respectively.

EC	DT	MLP	MBG of 1 DT	MBG of 1 MLP	MBG of 25 DT	MBG of 25 MLP	MBT of 25 DT	MBT of 25 MLP
TP	5.88	0.00	58.82	23.53	76.47	64.71	29.41	17.65
FP	0.00	0.00	23.30	32.95	20.17	32.95	1.99	15.63
TN	100	100	76.70	67.05	79.83	67.05	98.01	84.32
FN	94.12	100	41.18	76.47	23.53	35.29	70.59	82.35
Acc	95.66	95.39	75.88	65.04	79.67	66.94	94.85	81.30
Pre	100	$\propto$ (50)	71.63	41.66	79.12	66.63	93.66	50.03
Rec	5.88	0.00	58.82	23.53	76.47	64.71	29.41	17.65
FM	7.14	0.00	64.60	30.07	77.77	65.66	44.76	26.09

Another comparison between the performances of the two versions of the proposed method when using each of the two simple learners (i.e. decision tree and multilayer perceptron) as the base classifier is presented in the columns 5 and 6 of Table 1. These experiments show that the accuracy of the proposed method is acceptable when we use decision tree as base classifier. It will also show if the whole data points of dataset are used to construct the classifiers of the final ensemble, performance of the final ensemble may be still poor to identify the examples of the minority class. Table 1 depicts this important fact. As it is raised from Table 1, the use of the ensemble without applying the proposed method to balance the training data, does not solve the problem. However, applying the proposed method along with the use of ensemble significantly increases the efficiency. The last 2 columns (7 and 8) of Table 1 represent the results obtained by the ModifiedBoosting. As it is inferred from the Table 1, the algorithm can't compete with the the ModifiedBagging. It is worthy to mention that we slide the number from 1 to 25 and choose the value when the

F-measure hits its best. The comparison confirms why the ModifiedBagging outperforms the *EasyEnsemble* and *BalanceCascade*.

A schematic comparison between performances of the two mentioned versions of the proposed method is presented in Fig. 4. ROC curve of the proposed method using decision tree learner as the base classifier is superior to the one using multilayer perceptron learner as base classifier.

According to Fig. 4 sliding FP from 0 to 1, readers will find that if a better cut choice on FP axis is taken in ROC curve the results can even be improved. However, this is not stable because after a while increment in FP does cause improvement in TP. The above tests indicate that the accuracy of the proposed method outperforms the simple classifiers and some ensemble methods. The other conclusion is the superiority of the proposed method that uses decision tree as the base classifier rather than one that uses multilayer perceptron neural network as the base classifier.

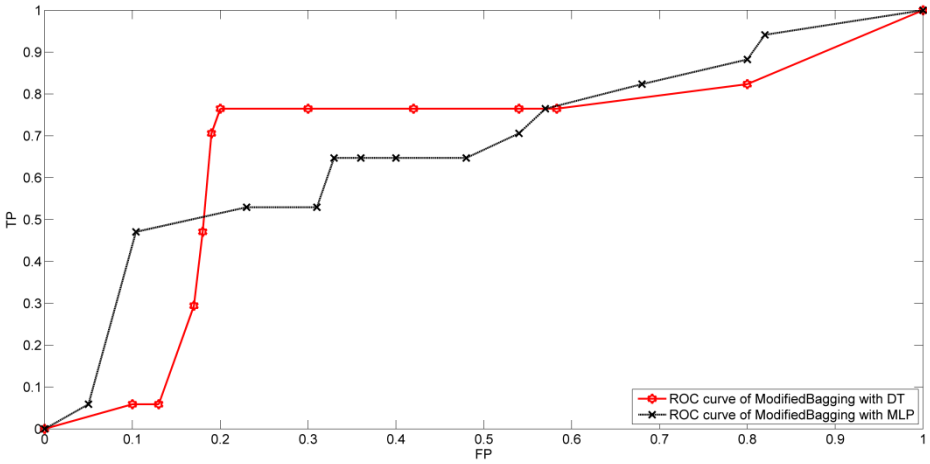

Fig. 4. ROC curve of the proposed ModifiedBagging with DT and MLP as base classifier

To demonstrate the efficacy of the proposed method in terms of the number of classifiers in the ensemble, please look at Fig. 5. In Fig. 5, the F-Measure of the proposed method in terms of the number of classifiers employed in the ensemble is depicted. All experimentations in Fig. 5 are averaged over 10 individual runs. The base classifier is chosen as decision tree. As it is inferred from Fig. 5, setting the number of classifiers to a value more than 23 does not affect much over F-Measure.

Now it is time to compare the proposed method with *EasyEnsemble* and *BalanceCascade* methods. By employing the two mentioned algorithms in the imbalanced dataset, any acceptable result is not again obtained according to Table 2. It is worthy to be mentioned that simple linear classifier used in reference [3] is used in both *EasyEnsemble* and *BalanceCascade* methods as base classifier. To reach the results of Table 2, *leave-one-out cross-validation* technique is used in all *EasyEnsemble*, *BalanceCascade* and *ModifiedBagging* methods. Comparing the proposed *ModifiedBagging* with *EasyEnsemble* and *BalanceCascade* in Table 2, we will reach the conclusion that the performances of the mentioned methods are weaker

than the proposed *ModifiedBagging* method. So it is concluded that it is not needed to go for reinforcement methods in such an imbalanced dataset. Considering the higher time orders of the mentioned algorithms to learn in such severely imbalanced datasets, we can claim that the proposed method in terms of both efficiency of learning and speed of learning is superior. In addition, we have generally proposed a framework to achieve a similar learning model in severely imbalanced datasets.

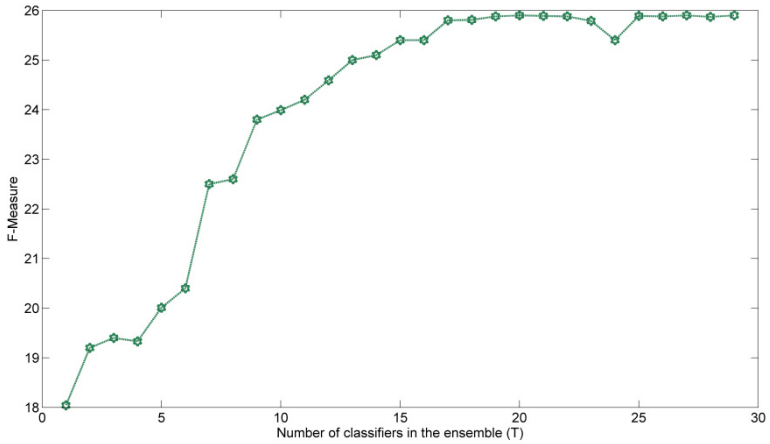

Fig. 5. F-Measure of ModifiedBagging with DT as base classifier in terms of number of classifiers

Perhaps the most important reason of failure in *EasyEnsemble* and *BalanceCascade* methods is hidden in severely imbalanced nature of the dataset. The reason of the well-performing the proposed method is its proper shape for learning small datasets. Consider when the data in the minority class is very low, the datasets created by *EasyEnsemble* and *BalanceCascade* are very low and consequently not suitable for learning of AdaBoost.

Table 2. Comparison of proposed method with EasyEnsemble and BalanceCascade methods

Evaluation Criterion	*EasyEnsemble* of 25 DTs	*BalanceCascade* 25 DTs	*ModifiedBagging* 25 DTs
TP	3/17=17.65	5/17=29.41	13/17=76.47
FP	31/352=8.81	43/352=12.22	71/352=20.17
TN	321/352=91.19	309/352=87.78	281/352=79.83
FN	14/17=82.35	12/17=70.59	4/17=23.53
Accuracy	324/369=87.80	314/369=85.09	294/369=79.67
Precision	66.70	70.44	79.13
Recall	17.65	29.41	76.47
F-Measure	27.91	41.50	77.78

5 Conclusions

In this paper a new method to learn in a severely imbalanced dataset where the number of data points in the minority class is very much less than the number of data points in the majority class is presented. This method is applied to an imbalanced breast cancer dataset. Inability of basic methods to learn in imbalanced spaces is also shown. Also due to the rare number of data points of the minority class in the benchmark, even the special-purpose methods are not able to learn the minority class. Inability of the special-purpose methods to learn the minority class in such severely imbalanced datasets guides us to present an innovative method fully suitable for these conditions. The main outcome of the research is in the field of medical research; to be used as a medical assistant. According to the profile and history of clients in the health centers, the proposed model can identify high risk clients in an automated manner. It can detect and treat an early breast cancer to prevent to use costly medical treatments and tests for clients and to help medical society to have an assistant.

References

[1] He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowledge And Data Engineering 21(9), 1263–1284 (2009)
[2] Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory Under Sampling for Class Imbalance Learning. In: Proc. Int'l Conf. Data Mining, pp. 965–969 (2006)
[3] Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory Under sampling for Class-Imbalance Learning. IEEE Transactions on Systems, Man, and Cybernetics-part B: Cybernetics (2009)
[4] Zhang, J., Mani, I.: KNN Approach to Imbalanced Data Distributions: A Case Study Involving Information Extraction. In: Int'l Conf. Machine Learning (2003)
[5] Hamzei, M., Kangavari, M.R.: Learning from imbalanced data. Technical Report, Iran University of Sci. & Tech., Iran (2010)
[6] Minaei, F., Soleimanian, M., Kheirkhah, D.: Investigation the relationship between risk factors of occurrence of breast tumor in women, Aranobidgol, Iran (2009)
[7] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-Sampling Technique. J. Artificial Intelligence Research 16, 321–357 (2002)
[8] He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In: Proc. Int'l J. Conf. Neural Networks, pp. 1322–1328 (2008)
[9] Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. ACM SIGKDD Explorations Newsletter 6(1), 20–29 (2004)
[10] Jo, T., Japkowicz, N.: Class Imbalances versus Small Disjuncts. ACM SIGKDD Explorations Newsletter 6(1), 40–49 (2004)
[11] Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: Improving Prediction of the Minority Class in Boosting. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 107–119. Springer, Heidelberg (2003)
[12] Schapire, R.E.: The strength of weak learn ability. Machine Learning 5(2), 1971–1227 (1990)